

## ⑫ 公開特許公報(A) 平4-205067

⑤Int.Cl.<sup>5</sup>

識別記号

庁内整理番号

⑬公開 平成4年(1992)7月27日

G 06 F 15/62

3 4 0

8125-5L

審査請求 未請求 請求項の数 1 (全8頁)

⑭発明の名称 アニメーション自動生成方式

⑰特 願 平2-321540

⑱出 願 平2(1990)11月26日

⑲発 明 者 藤 田 卓 志 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内⑲発 明 者 林 一 司 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内⑲発 明 者 松 本 智 佳 子 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内

⑳出 願 人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

㉑代 理 人 弁理士 岡田 守弘

## 明 細 書

## 1. 発明の名称

アニメーション自動生成方式

## 2. 特許請求の範囲

アニメーションを自動生成するアニメーション  
自動生成方式において、キャラクタの状態推移を表現するための任意個  
の関数名に対応づけて任意個の変数名を予め定義  
する状態推移関数リスト(2)と、この状態推移関数リスト(2)で定義した関数名に  
対応する関数を予め作成して格納する状態推移関  
数ライブラリ(3)と、上記状態推移関数リスト(2)で定義した変数名に  
対応する状態変数(外部状態変数、内部状態変数、  
参照変数)の値を格納する状態変数リスト(4)とを  
持つキャラクタ動作生成部(1)を複数設け、共有メ  
モリ(6)などを介して相互に通信を行うように構成  
し、実行指示に対応して、上記キャラクタ動作生成  
部(1)が上記共有メモリ(6)などから参照変数の値を  
読み込むと共に上記状態推移関数リスト(2)から関  
数名およびその変数を1組取り出し、上記状態推  
移関数ライブラリ(3)中から起動した当該関数名の  
関数が上記状態変数リスト(4)中の更新前の状態変  
数の値を読み込んで更新後の状態変数の値を求め  
ることを繰り返すまで繰り返した後、外部状  
態変数の値を共有メモリ(6)などにコピーするよう  
に構成したことを特徴とするアニメーション自動  
生成方式。

## 3. 発明の詳細な説明

〔概要〕

アニメーションを自動生成するアニメーション  
自動生成方式に関し、複数のキャラクタが相互に影響を及ぼし合いな  
がらリアルタイムにアニメーションを自動生成す  
ると共に分散処理を可能にすることを目的とし、

キャラクタの状態推移を表現するための任意個

の関数名に対応づけて任意個の変数名を予め定義する状態推移関数リストと、この状態推移関数リストで定義した関数名に対応する関数を予め作成して格納する状態推移関数ライブラリと、上記状態推移関数リストで定義した変数名に対応する状態変数（外部状態変数、内部状態変数、参照変数）の値を格納する状態変数リストとを持つキャラクター動作生成部を複数設け、共有メモリなどを介して相互に通信を行うように構成し、実行指示に対応して、上記キャラクター動作生成部が上記共有メモリなどから参照変数の値を読み込むと共に上記状態推移関数リストから関数名およびその変数を1組取り出し、上記状態推移関数ライブラリ中から起動した当該関数名の関数が上記状態変数リスト中の更新前の状態変数の値を読み込んで更新後の状態変数の値を求めることを繰り返すまで繰り返した後、外部状態変数の値を共有メモリなどにコピーするように構成する。

〔産業上の利用分野〕

〔発明が解決しようとする課題〕

上述した従来の方式でアニメーションを生成する場合、まずストーリーを設定し、これに従ったキャラクターの動きを全て人が指定する必要がある、動きのデータの再利用が難しく、生産性が極めて悪いという問題があった。このため、シミュレーションによるキャラクターの動きの多くの場合に対応するために、複数のキャラクターが相互に影響するようなモデルに関する汎用的な動作生成の方式が要求されている。また、対話的アニメーションシステムや仮想体験システムでは、リアルタイムのシミュレーション実行が必要となり、そのために並列処理などによる負荷分散に適した方法が要求されている。

本発明は、複数のキャラクターが相互に影響を及ぼし合いながらリアルタイムにアニメーションを自動生成すると共に分散処理を可能にすることを目的としている。

〔課題を解決するための手段〕

本発明は、アニメーションを自動生成するアニメーション自動生成方式であって、外部からの入力にリアルタイムにตอบสนองしたアニメーションを生成するアニメーション自動生成方式に関するものである。特に、計算機上に構築した仮想的な世界モデルのシミュレーションを行ってリアルタイムにアニメーションを生成して表示し、擬似的に体験できるようにすることが望まれている。

〔従来の技術〕

従来からのコンピュータアニメーションの制作の主流は、キーフレームアニメーションである。これは、アニメーションの系列のうち、飛び飛びのフレーム（キーフレーム）のデザインを行い、これらの間のフレームの映像を中割により生成して表示する方式であり、予め全てのキャラクターの動きが決まっているアニメーションを制作する際のアニメータの負担を軽減することを目的としていた。

第1図を参照した課題を解決するための手段を説明する。

第1図において、状態推移関数リスト2は、キャラクターの状態推移を表現するための任意個の関数名に対応づけて任意個の変数名を予め定義したリストである。

状態推移関数ライブラリ3は、関数名に対応する関数を予め作成して格納するライブラリである。

状態変数リスト4は、変数名に対応する状態変数（外部状態変数、内部状態変数、参照変数）の値を格納するリストである。

キャラクター動作生成部1は、関数の実行および状態変数（外部状態変数、内部状態変数、参照変数）の更新などを行うものである。

共有メモリ6は、状態変数（外部状態変数など）を格納するものである。

〔作用〕

本発明は、第1図に示すように、実行指示に対応して、キャラクター動作生成部1が共有メモリ6

などから参照変数を読み込んで保持すると共に状態推移関数リスト2から関数名およびその変数を1組取り出し、状態推移関数ライブラリ3中から起動した当該関数名の関数が状態変数リスト4中の更新前の状態変数の値を読み込んで更新後の状態変数の値を求めることを組が無くなるまで繰り返した後、外部状態変数の値を共有メモリ6などにコピーするようにしている。

従って、複数組のキャラクタ動作生成部1が共有メモリ6などにコピーされている外部状態変数のうちから参照変数について内部に読み込んで自己のキャラクタの状態推移をそれぞれ求めて更新すると共に外部状態変数を共有メモリ6などにコピーすることを繰り返すことにより、複数のキャラクタが相互に影響を及ぼし合いながらリアルタイムにアニメーションを自動生成することが可能となると共に分散処理を行うことが可能となる。

(実施例)

次に、第1図から第7図を用いて本発明の実施

例の構成および動作を順次詳細に説明する。

第1図において、キャラクタ動作生成部1は、状態推移関数リスト2、状態推移関数ライブラリ3、および更新前の状態変数リスト4を格納する更新前バッファ、更新後の状態変数リスト4を格納する更新後バッファなどから構成され、キャラクタの外見的特徴を表す関数名の関数の実行を行い、共有メモリ6から参照変数を読み込んで更新前バッファに格納されている状態変数リスト4のうちの該当する状態変数(外部状態変数、内部状態変数、参照変数)の値を計算して更新後バッファに格納されている状態変数リスト4のうちの該当する状態変数(外部状態変数、内部状態変数、参照変数)の値を更新したり、この更新した外部状態変数の値を共有メモリ6にコピーしたりなどするものである。

状態推移関数リスト2は、キャラクタの状態推移を表現するための任意個の関数名(図中ではcalc position)に対応づけて任意個の変数名(図中ではposition、velocity、dt)を予め定義したり

ストである。

状態推移関数ライブラリ3は、状態推移関数リスト2で予め定義した関数名(図中ではcalc position)に対応する関数(例えば第4図calc positionに示すような処理)を予め作成して格納するライブラリである。

状態変数リスト4は、状態推移関数リスト2で関数名に対応づけて予め定義した変数名(引数)に対応する状態変数(外部状態変数、内部状態変数、参照変数、第5図参照)の値を格納するリストである。

通信部5は、他のキャラクタ動作生成部1、入出力部7との間でデータの授受を行う共有メモリ6などである。ここでは、各キャラクタ動作生成部1が他のキャラクタに影響を与える外部状態変数などの値を格納したり、自己が影響を受ける参照変数の値を読み込んだりするためのものである。

入出力部7は、位置センサ、音声センサなどからなる入力部8およびキャラクタを表示する画像生成・表示部9などから構成されるものである。

第2図は、本発明のシステム構成図を示す。

第2図において、キャラクタ動作生成部1は、関数の実行および状態変数の更新を行う処理部1-1、および外部状態変数、内部状態変数、参照変数、状態推移関数などを格納する局所メモリ1-2などから構成されている。これら複数のキャラクタ動作生成部1は、共有メモリ6を介してデータの授受(外部状態変数の値を共有メモリ6にコピー、共有メモリ6から参照変数の値を読み込むなどしてデータの授受)を行い、評価時間間隔毎に順次キャラクタについて相互の影響を及ぼしながら動くようなアニメーションを自動生成するようにしている。

入力装置7-1は、位置センサ、音声センサなどの入力装置である。

入力認識部7-2は、各種センサからの入力を認識し、共有メモリ6に書き込み、キャラクタ動作生成部1に通知したりなどするものである。

画像生成部9-1は、共有メモリ6上に書き込まれた外部状態変数などの値(重心位置、色、サ

イズなどの値)をもとに各キャラクターの画像を生成するものである。

表示装置9-2は、ディスプレイ装置であって、画像生成部9-1によって生成された各キャラクターの画像を表示するものである。

次に、第3図ないし第7図を用いて、第1図、第2図構成の動作を順次具体的に説明する。

まず、第3図アニメーションデザイン手順例に従って、第1図状態推移関数リスト2、状態推移関数ライブラリ3、および状態変数リスト4の作成を行う。

第3図において、①は、状態変数リスト4の作成を行う。これは、例えば第6図に示すように、アニメーションを構成するキャラクター名"Whale(クジラ)"について、外部状態変数、内部状態変数、参照変数を作成する(第6図を用いて後述する)。

②は、状態推移関数ライブラリを作成する。これは、例えば第5図に示すように、アニメーションを構成するキャラクター名"Whale(クジラ)"の関数名"calc-position"の関数をライブラリとして

作成する(第5図を用いて後述する)。

③は、状態推移関数リストを作成する。これは、例えば第4図に示すように、アニメーションを構成するキャラクター名"Whale(クジラ)"について、関数名に対応づけて変数1、変数2などを定義して作成する(第4図を用いて後述する)。

以上のように、状態変数リスト4、状態推移関数ライブラリ3、状態推移関数リスト2を作成することにより、本発明のアニメーションデザインを行ったこととなる。

次に、これら作成した状態推移関数リスト2、状態推移関数ライブラリ3、状態変数リスト4について順次詳細に説明する。

第4図は、状態推移関数リスト例を示す。これは、第1図画像生成・表示部9上に表示したアニメーションを構成する複数のキャラクターのうちのキャラクター名"Whale(クジラ)"について、外見のおよび内的状態の変化を表現するための関数名を図示のように予め定義したものである。具体的に説明すると、例えば

```
calc-position position velocity dt・・・(4)
```

の、calc-position は関数名を表し、position、velocity、dtが変数名をそれぞれ表す。

第5図は、状態推移関数ライブラリ例を示す。これは、第2図処理部1-1が実行するための関数であって、図中では関数名"calc-position"の関数である。図示のような処理によって位置変化の計算、状態変数の更新などを行うようにしている(第7図を用いて後述する)。

第6図は、状態変数リスト例を示す。これは、キャラクター名"Whale"の外部状態変数、内部状態変数、参照変数について作成したものである。ここで、

外部状態変数：キャラクター名"Whale"の位置や速度、質量、形などの外部状態を表す変数であって、図中では位置"position"、"velocity"などが作成されている。従って、当該キャラクター名"Whale"は変数名"position"、"velocity"など(図示してないがこの他に色、サイズなどもある)に従って評価時刻の間隔で移動などすることとなる。

具体的に説明すると、

```
global static-float3 position 10.0 20.0 45.0
0・・・・・・・・・(1)
```

の、globalは外部状態変数を表し、static-float3は型名であって保存的浮動小数点3次元ベクトル変数を表し、positionは変数名であってWhaleの重心位置を表し、10.0 20.0 45.0はX、Y、Zの座標位置をそれぞれ表す。

内部状態変数：キャラクター名"Whale"の空腹度や疲労度などの体の状態あるいは嬉しさや寂しさなどの精神的状態などのように、外部から直接観察できない内部状態を表す変数であって、図中では"appetite(食欲)"、"fatigue(身体の疲労)"などが作成されている。従って、当該キャラクター名"Whale"の内部状態変数の変数名"appetite"、"fatigue"などに従って評価時刻の間隔で状態推移関数ライブラリ3中の該当する関数が計算を行い、外部状態変数に影響を与えるようにしている。また、記述を具体的に説明すると、

```
local static-float appetite 0.7・・・・・・(2)
```

の、local は内部状態変数を表し、static-float は型名であって保存的浮動小数点スカラー変数を表し、appetiteは変数名であってWhale の空腹度を表し、0.70は空腹度の値を表す。

参照変数：自分のキャラクタ名"Whale" の状態変化を求める際に必要な状態変数を計算に先立って読み込むためのものであって、図中では参照キャラクタ名"A1"の変数名"position"を、変数名"pos-A1"として読み込むことなどが作成されている。従って、当該キャラクタ名"Whale" の参照変数に伴って評価時刻の間隔で共有メモリ6から自己の計算に必要なデータを予め取り込んで内部に保持するようにしている。また、記述を具体的に説明すると、

refer const-float3 pos-A1 A1 position · (3)  
の、refer は参照変数を表し、const-float3は読出専用浮動小数点3次元ベクトル変数を表し、pos-A1は変数名を表し、A1は参照キャラクタ名を表し、positionは参照キャラクタ名の変数名を表す。  
次に、第7図生成例の順序に従い、第1図、第

④は、更新後の状態変数の値を求める。これは、例えば④で起動した関数"calc-position" が状態変数（参照変数、内部状態変数など）の値をもとに更新後の状態変数の値を求める。

⑤は、結果を、更新後の状態変数のバッファに書き込む。これは、例えば④で起動した関数"calc-position" が、⑤で求めた更新後の状態変数の値について、更新後のバッファに書き込む。

⑥は、状態推移関数リストが終わりか否かを判別する。YES の場合には、⑥で更新後のバッファの内容を、更新前のバッファにコピーし、⑥で外部状態変数の値を共有メモリにコピーし、⑥に進む。一方、NO の場合には、次の組について、④以降を繰り返し行う。

⑦は、アニメーションが終了か否かを判別する。YES の場合には、終了する。NO の場合には、④以降で次の評価時刻についてアニメーションを行う。

以上の手順によって、第3図アニメーションデザイン手順によって予め作成された状態推移関数

2図構成の動作を詳細に説明する。

第7図において、④は、時刻を読み込む。これは、状態変数の変化量の計算に必要な時間情報として、システム時計から絶対時刻を読み込み、評価の対象となる時刻を特定する。更に、前評価時刻との時間差を求める。

⑤は、共有メモリ6から、他のキャラクタの外部状態変数のコピーを参照変数に読み込む。これは、第2図で共有メモリ6からキャラクタ動作生成部1が自己の参照変数にコピーを読み込む。

⑥は、状態推移関数リスト2より、関数とその引数（状態変数）の組を1つ取り出す。例えば第1図関数"calc-position" と状態変数"position"の組を取り出す。

⑦は、関数を起動し、更新前の状態変数を読み込む。これは、例えば⑥で取り出した組の関数"calc-position" を状態推移関数ライブラリ3から取り出して起動し、この起動した関数"calc-position" が状態変数"position"について、更新前バッファからその値を読み込む。

リスト2、状態推移関数ライブラリ3、および状態変数リスト4をもとに、キャラクタが相互に影響を及ぼし合いながら、アニメーションを自動作成することが可能となる。

〔発明の効果〕

以上説明したように、本発明によれば、予め作成した状態推移関数リスト2、状態推移関数ライブラリ3および状態変数リスト4をもとに、複数のキャラクタ動作生成部1が共有メモリ6などにコピーされている外部状態変数のうちから参照変数について内部に読み込んで自己のキャラクタの状態推移をそれぞれ求めて更新すると共に外部状態変数を共有メモリ6などにコピーすることによってアニメーションを作成する構成を採用しているため、複数のキャラクタが相互に影響を及ぼし合いながらリアルタイムにアニメーションを自動生成することができると共に分散処理によってリアルタイムに高速に生成することができる。これにより、ストーリー自体をシステムが生成

しながらキャラクタの状態推移を生成し、アニメーションを生成するための汎用的システムを実現できた。また、アニメーション制作において、キャラクタの状態推移規則を直接利用できるのも、従来のようにキャラクタの動きを全て人手で指定する手法に比べて、データの再利用が容易となり、アニメーション制作の効率を向上させることができる。更に、対話的アニメーションシステムや疑似体験システムの実現のための汎用的な手法としても利用できる。

4. 図面の簡単な説明

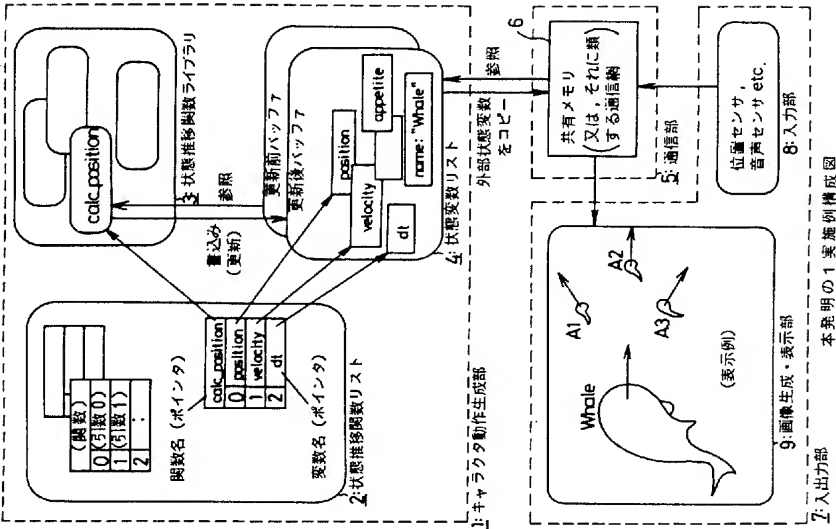
- 第1図は本発明の1実施例構成図
- 第2図は本発明のシステム構成図
- 第3図は本発明のアニメーションデザイン手順例
- 第4図は本発明の状態推移関数リスト例
- 第5図は本発明の状態推移関数ライブラリ例
- 第6図は本発明の状態変数リスト例
- 第7図は本発明のキャラクタの動作生成例

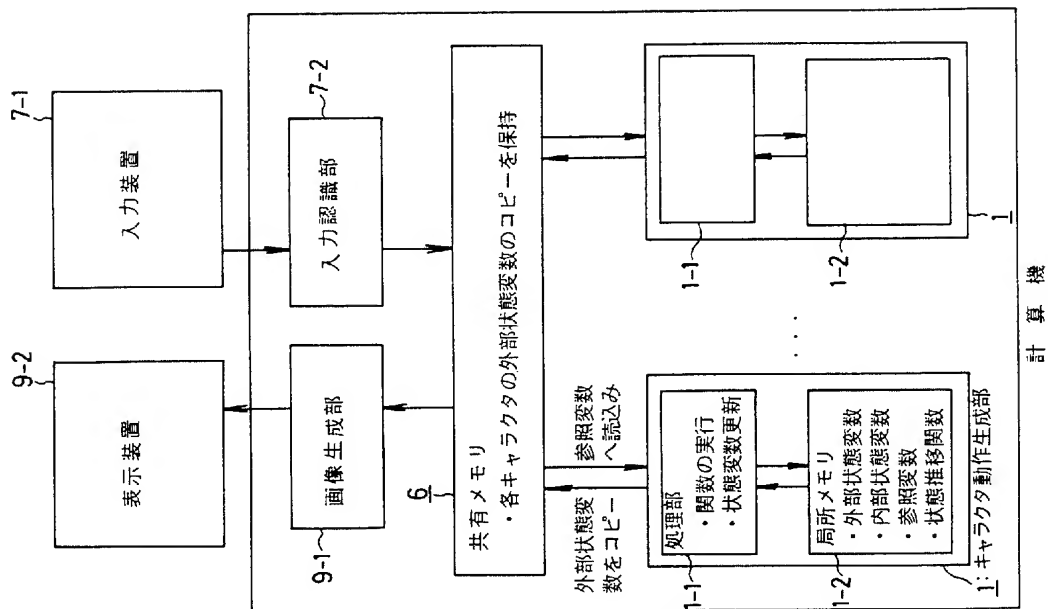
を示す。

図中、1：キャラクタ動作生成部

- 1-1：処理部
- 1-2：局所メモリ
- 2：状態推移関数リスト
- 3：状態推移関数ライブラリ
- 4：状態変数リスト
- 5：通信部
- 6：共有メモリ
- 7：入出力部
- 8：入力部
- 9：画像生成・表示部
- 9-1：画像生成部
- 9-2：表示装置

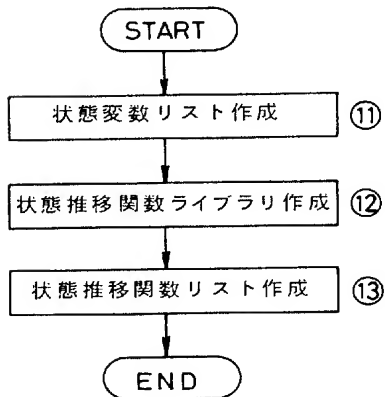
特許出願人 富士通株式会社  
代理人弁理士 岡田 守弘





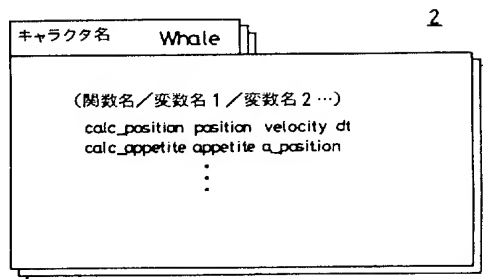
本発明のシステム構成図

第 2 図



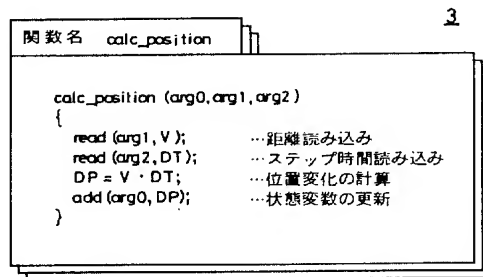
本発明のアニメーションデザイン手順例

第 3 図



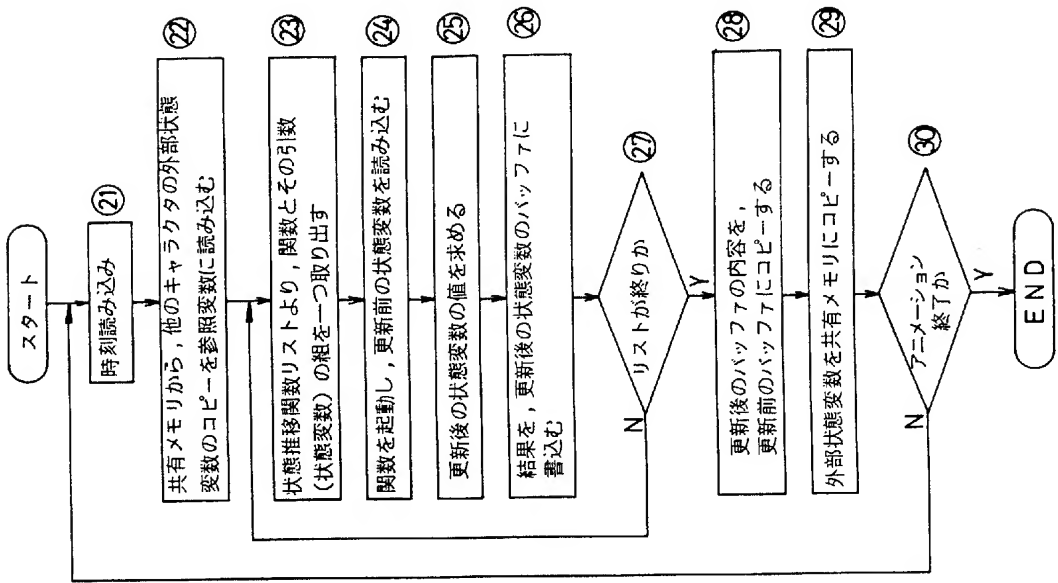
本発明の状態推移関数リスト例

第 4 図

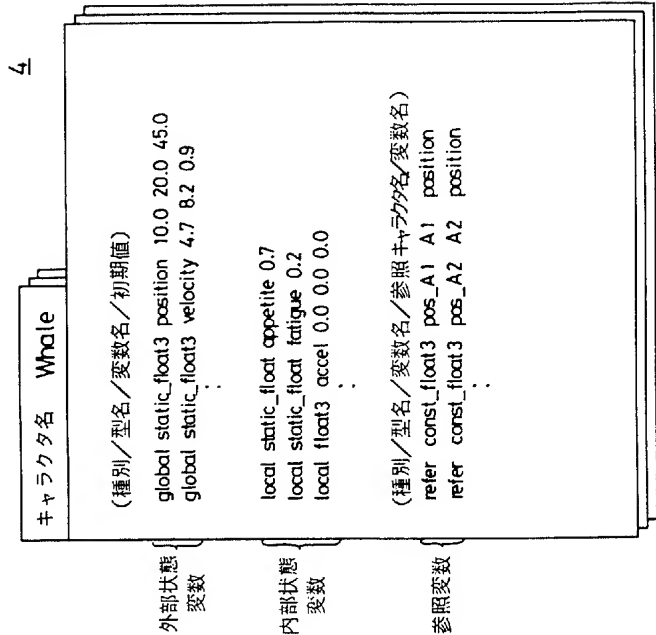


本発明の状態推移関数ライブラリ例

第 5 図



本発明のキャラクターの動作生成例



種別

global : 外部状態変数  
local : 内部状態変数  
refer : 参照変数

型名 (例)

static\_float3 : 保存的浮動小数点 3 次元ベクトル変数  
static\_float : 保存的浮動小数点スカラ変数  
float3 : 非保存的浮動小数点 3 次元ベクトル変数  
const\_float3 : 読出専用浮動小数点 3 次元ベクトル変数

本発明の状態変数リスト例



**PAT-NO:** JP404205067A  
**DOCUMENT-  
IDENTIFIER:** JP 04205067 A  
**TITLE:** AUTOMATIC ANIMATION  
GENERATING METHOD  
**PUBN-DATE:** July 27, 1992

**INVENTOR-INFORMATION:**

<b>NAME</b>	<b>COUNTRY</b>
FUJITA, TAKUSHI	
HAYASHI, ICHIIJI	
MATSUMOTO, CHIKAKO	

**ASSIGNEE-INFORMATION:**

<b>NAME</b>	<b>COUNTRY</b>
FUJITSU LTD	N/A

**APPL-NO:** JP02321540  
**APPL-DATE:** November 26, 1990

**INT-CL (IPC):** G06F015/62

**ABSTRACT:**

PURPOSE: To make it easy to reuse data

and to improve the efficiency of producing an animation by generating the transition of character state and repeating copying the values of external state variables in a common memory.

CONSTITUTION: A plurality of sets of character operation generating sections 1 send and receive data via a common memory 6, respectively obtain and update state transitions of own characters sequentially at every evaluation time interval. By repeating copying external state variables in a common memory 6 or the like, while a plural characters mutually exert an influence, the animation is automatically produced in a realtime.

COPYRIGHT: (C)1992,JPO&Japio